

36장. 디스트럭처링 할당

1. **디스트럭처링 할당**(구조 분해 할당)은 구조화된 배열과 같은 이터러블 또는 객체를 destructuring (비구조화, 구조 파괴)하여 1개 이상 변수에 개별적으로 할당하는 것을 의미.

36.1 배열 디스트럭처링 할당

1. ES5에서 구조화된 배열을 디스트럭처링하여 1개 이상 변수에 할당하는 방법

```
// ES5
var arr = [1, 2, 3];

var one   = arr[0];
var two   = arr[1];
var three = arr[2];

console.log(one, two, three); // 1 2 3
```

2. ES6에서는 배열 디스트럭처링 할당의 대상(할당문의 우변)은 이터러블이어야 하며, 할당 기준은 배열의 인덱스

```
const arr = [1, 2, 3];

// ES6 배열 디스트럭처링 할당
// 변수 one, two, three를 선언하고 배열 arr을 디스트럭처링하여 할당한다.
// 이때 할당 기준은 배열의 인덱스다.
const [one, two, three] = arr;

console.log(one, two, three); // 1 2 3
```

- 순서대로 할당된다. 변수의 개수와 이터러블의 요소 개수가 반드시 일치할 필요는 X

➡ 배열과 같은 이터러블에서 필요한 요소만 추출하여 변수에 할당하고 싶을 때 유용.

36.2 객체 디스트럭처링 할당

1. ES5에서 객체의 각 프로퍼티를 객체로부터 디스트럭처링하여 변수에 할당하기 위해서는, 프로퍼티 키를 사용해야 한다.

```
// ES5
var user = { firstName: 'Ungmo', lastName: 'Lee' };
```

```
var firstName = user.firstName;
var lastName = user.lastName;

console.log(firstName, lastName); // Ungmo Lee
```

2. ES6의 객체 디스트럭처링 할당의 대상(할당문의 우변)은 객체이어야 하며, 할당 기준은 프로퍼티 키다. 즉, 순서는 의미가 없으며 선언된 변수 이름과 프로퍼티 키가 일치하면 할당된다.

```
const user = { firstName: 'Ungmo', lastName: 'Lee' };

// ES6 객체 디스트럭처링 할당
// 변수 lastName, firstName을 선언하고 user 객체를 디스트럭처링하여 할당한다.
// 이때 프로퍼티 키를 기준으로 디스트럭처링 할당이 이루어진다. 순서는 의미가 없다.
const { lastName, firstName } = user;

console.log(firstName, lastName); // Ungmo Lee
```

3. 객체의 프로퍼티 키와 다른 변수 이름으로 프로퍼티 값을 할당받으려면 다음과 같이 변수 선언

```
const user = { firstName: 'Ungmo', lastName: 'Lee' };

// 프로퍼티 키를 기준으로 디스트럭처링 할당이 이루어진다.
// 프로퍼티 키가 lastName인 프로퍼티 값을 ln에 할당하고,
// 프로퍼티 키가 firstName인 프로퍼티 값을 fn에 할당한다.
const { lastName: ln, firstName: fn } = user;

console.log(fn, ln); // Ungmo Lee
```

4. 객체 디스트럭처링 할당은 객체에서 프로퍼티 키로 필요한 프로퍼티 값만 추출하여 변수에 할당하고 싶을 때 유용하다.

5. 배열의 요소가 객체인 경우,

배열 디스트럭처링 할당과 객체 디스트럭처링 할당을 혼용할 수 있다.

```
const todos = [
  { id: 1, content: 'HTML', completed: true },
  { id: 2, content: 'CSS', completed: false },
  { id: 3, content: 'JS', completed: false }
];

// todos 배열의 두 번째 요소인 객체로부터 id 프로퍼티만 추출한다.
```

```
const [, { id }] = todos;  
console.log(id); // 2
```